ReactOS Build Environment

Version 2.3 for Unix-based operating systems

Base package with i386 and amd64 compilers

Introduction

The ReactOS Build Environment ("RosBE") is the official build environment for compiling ReactOS. This version is targeted at Unix-based operating systems such as Linux, FreeBSD or macOS.

The installation process is divided into two parts: First a script builds and installs the needed compiler tools for your operating system. Therefore you need the "gcc" compiler for your system and some other build tools.

Afterwards, you can create a desktop shortcut using another script. This shortcut opens a Terminal window and prepares all environment settings for building ReactOS.

Needed Applications and Libraries

Make sure that the following applications and libraries are installed on your system. They are needed to build the ReactOS Build Environment. The "RosBE-Builder.sh" script automatically checks for their presence:

- GNU Bash
  The executable file and the package name are often called "bash".
  All scripts are exclusively designed for the Bash shell.

- GNU GCC
  You need a GCC compiler that is suitable for your system. The executable file and the package name are often called "gcc".
  Your GCC compiler needs to be able to compile both C and C++ sources, hence you also need to have the "g++" module installed.
  Make sure you have GCC 7.x or later installed, as earlier versions have not been tested.

- Grep
  The executable file and the package name are often called "grep".

- GNU Make
  Make sure that you have the GNU version of "make" installed. Operating systems like FreeBSD have a different version installed by default.
  The executable file and the package name are often called "make" or "gmake".

- GNU Makeinfo
  The executable file is often called "makeinfo". This file is usually in a package called "texinfo".

- zlib
  The library file is generally called "libz.so". Package names vary, one example for Debian-based Linux distributions is "zlib1g-dev".

Installation of the Base package and i386 compiler

The Installation of RosBE is very simple as there are scripts that do most of the work for you. Follow these steps to compile and install RosBE for your operating system:

1. Open a Terminal window and run "sudo ./RosBE-Builder.sh" in the directory of the unpacked RosBE files.
   NOTE: Becoming a root user through "sudo" is only required for installing RosBE into the default directory ("/usr/local/RosBE"). If you want to install it into a personal directory, you don't need superuser privileges.
   Advanced users can also make use of CC, CFLAGS, CXX, and CXXFLAGS variables to modulate the way tools will be built.

2. Follow the directions on the screen until the script ends with "Finished successfully!"

3. Call the "createshortcut.sh" script in the installation directory (usually "/usr/local/RosBE") to create a desktop shortcut for the Build Environment.

Afterwards, you can easily open a Build Environment for ReactOS by clicking the shortcut.

Installation of the amd64 compiler

The amd64 compiler can only be installed after you have installed the Base package and i386 compiler.

When this is done, just run "sudo ./RosBE-Builder-amd64.sh" and follow the instructions. The same options and variables can be used as for the base installation script.

Uninstallation

If you want to uninstall RosBE, just delete the entire installation directory and any shortcuts you have created.

Version Information

This version of RosBE for Unix-based operating systems uses the following components:

- GNU Binutils v2.36.1
  compiled for i686-w64-mingw32
  patched with:

  - https://raw.githubusercontent.com/reactos/RosBE/f9e43c3b01dcbd507e5b1454f654f4493955b131/Patches/binutils-2.36.1-jerome-gardou-optional-code-argument.patch

- GCC, The GNU Compiler Collection v8.4.0
  compiled for i686-w64-mingw32

> **NOTE:** This GCC version only supports C and C++

- Bison 3.5.4
  patched with:

  - https://raw.githubusercontent.com/reactos/RosBE/e87b00c8f8732ed3fa393b9b05a12093ae5942e8/Patches/bison-3.5-reactos-fix-win32-build.patch

- CMake 3.17.2-ReactOS
  (revision 07c58033d74bad5466775fb2dad7d84494a846bd from https://github.com/reactos/CMake/tree/cmake-3.17.2-reactos)

- Flex 2.6.4+ (revision 8b1fbf674f2e038df9cf1fe7725617e3837ae2a9)

- GMP 6.2.0
  patched with:

  - https://raw.githubusercontent.com/reactos/RosBE/e87b00c8f8732ed3fa393b9b05a12093ae5942e8/Patches/GMP-6.2.0-C89-fixes.patch

- Mingw-w64 6.0.0

- MPC 1.1.0

- MPFR 4.0.2

- Ninja 1.10.2-ReactOS
  (revision b661be7af3f98bd9a29caa7b35702bd5248f8ed8 from https://github.com/reactos/ninja/tree/ninja-1.10.2-reactos)

Tested Operating Systems

The Build Environment has been designed to work on all Unix-based operating systems. Nevertheless, it is impossible to test it on all possible systems out there.

The current primary development and test system is Ubuntu 20.04. But users also reported that it works well under other Linux systems such as Fedora or openSUSE and Unix derivates like FreeBSD or macOS.

You should be able to compile the build tools on both i386 (32-bit) and x86_64 (64-bit) systems.

Websites

Binutils, the Binary Tools Collection

https://www.gnu.org/software/binutils

Bison, general-purpose parser generator

https://www.gnu.org/software/bison

CMake, Cross Platform Make

https://cmake.org

Flex, The Fast Lexical Analyzer

https://github.com/westes/flex

GCC, the GNU Compiler Collection

https://gcc.gnu.org

GMP, the GNU MP Bignum library

https://gmplib.org

The GNU MPFR library

http://www.mpfr.org

Mingw-w64

https://mingw-w64.org

Ninja, a small build system with a focus on speed

https://ninja-build.org

Credits

Colin Finck (colin@reactos.org)

Pierre Schweitzer (pierre@reactos.org)

Victor Perevertkin (victor.perevertkin@reactos.org)

Thanks go also to the members of the RosBE-Windows team, namely:

Daniel Reimer (daniel.reimer@reactos.org)

Peter Ward (dralnix@gmail.com)

Special thanks to ROSFan, who created the nice ReactOS Build Environment Logo.

Special thanks also goes out to all those who provide so much help by testing out new versions, you know who you are. Without you, RosBE would not be what it is.

Credits go to Gregor Anich who created the original ReactOS Build Environment v0.2-3.4.4.

Inspired by Dazzle from TinyKRNL.